REMARKS

By this Response, the Applicants hereby amend the claims by adding new claims 31-48. Support for the new claims is found in the specification as originally filed. No new matter has been added. Claims 1-48 are presently pending in this application. Reconsideration of this application for allowance of all pending claims are hereby respectfully requested in view of the amendments to the claims and the following remarks.

Objection based on informality

In Section 1 of the Office Action, dated May 9, 2005, the Examiner objected to the disclosure based on informality of the specification. Paragraph 111 of the specification has been amended to address the objection.

Rejection under 35 U.S.C. § 112

In Section 2 of the Office Action, dated May 9, 2005, the Examiner rejected claims 1-30 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. First, the Examiner indicated that the claim language "to account for an elemental width of the data" is unclear. The Applicants respectfully point out that the claim language "to account for" has ordinary meaning of "a basis or ground". Therefore, the claimed feature "dynamically partition data received from the data path to account for an elemental width of the data" can be understood to mean that dynamically partitioning data received from the data path is performed based on an elemental width of the data.

In addition, the Examiner questioned the relationship between "operand registers" and the "at least one register file". The Applicants respectfully respond that operand registers, as recited in claim 1, are not necessarily within the "at least one register file".

The Examiner's rejections under 35 U.S.C. § 112, second paragraph, have been traversed. The Applicants hereby respectfully request that the rejection under 35 U.S.C. § 112, second paragraph, be withdrawn.

Rejection under 35 U.S.C. § 102

In Section 4 of the Office Action, claims 1-30 have been rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 5,887,183 issued to Agarwal et al. (hereafter "Agarwal"). The Applicants respectfully traverse the rejection.

Agarwal teaches a method and system to interface between a memory and a plurality of processing elements in a Single Instruction Multiple Data (SIMD) parallel processing environment (see Figs. 2 and 3). It is commonly known that a SIMD machine or execution unit has a plurality of processing elements that can process data simultaneously. For example, in a SIMD execution unit having 64 processing elements, each of the 64 processing elements may perform computation on one element of an 8x8 (64 data elements) matrix. To perform such parallel processing in one instruction, data elements to be processed by individual processing elements need to be dispatched to appropriate processing elements prior to execution. This is a loading process. In addition, once the processing is completed, results produced by individual processing elements need to be gathered and then stored in a memory. This is a store process. Agarwal discloses a method and system for loading and storing vectors in a plurality of modes (see Col. 5, lines 11-14).

Specifically, Agarwal discloses how data in a memory stored in one pattern can be loaded into multiple processing elements of a SIMD execution unit in one of several different patterns. For example, data stored sequentially in memory can be loaded into SIMD according to stride-1, stride-(-1), stride-2, ... stride-n patterns (Figs. 4A, 5A, 6A, and 7A). The mapping from a memory pattern to one of the several patterns in the SIMD execution unit during loading is accomplished through a load data routing logic 258 during a loading process (see col. 10, lines 13-17, lines 65-67, col. 11, lines 40-42, col. 12, lines 23-26). In addition, Agarwal teaches how data elements in different processing elements, organized in one of the above mentioned several patterns, can be transported and stored back into a memory (see Figs. 4B, 5B, 6B, and 7B). The mapping from one of the several patterns in the SIMD execution unit to a memory storage pattern during storing is accomplished through a store data routing logic 258 during a store process (see col. 10, lines 43-46, col. 11, lines 11-17, lines 60-65, col. 12, lines 35-39).

The load data routing logic 258 and the store data routing logic 259, according to Agarwal, are capable of routing data according to an original pattern and a target pattern. For example, data in memory organized in a row-oriented manner may be mapped by the load data routing logic 258 to vector registers in SIMD in a column-oriented manner. This is achieved, according to Agarwal, through stride-n operation by routing each piece of data from the memory to a target vector register in the SIMD (see Fig. 7A). Store process is an reverse operation of the load operation. Although it is possible to perform data alignment and single-precision / double-precision data selection during load/store operations (see col. 15, lines 31-62), Agarwal, as disclosed, perform mainly routing on each piece of data as a unit (as the names "load data routing logic" and "data store routing logic" imply).

Claim 1 of the present invention recites "a multi-precision execution unit ... configurable to dynamically partition data received from the data path to account for an elemental width ...

wherein the elemental width of the data is equal or narrower than the data path, the multi-precision execution unit being capable of performing group floating-point operations on multiple operands in partitioned fields of operand registers ...". That is, the recited multi-precision execution unit has features including (1) capable of dividing data from the data path width-wise into a variable number of data elements corresponding to partitioned fields, (2) that the division (or partition) is based upon the size of the data elements or the elemental width which is equal or narrower than the data path, and (3) that the partitioned fields correspond to multiple operands that are used in group floating-point operations.

Agarwal does not teach, disclose, or suggest that data be partitioned into a variable number of data elements no wider than the data path based on an elemental width corresponding to the size of the data elements. Although Agarwal discloses obtaining single-precision and double-precision data elements, "double-precision data elements are forwarded straight through" (see col. 15, lines 46-47) and "single-precision operands must be expanded so that each 32-bit element is placed at every 32-bit location in the data flow, padding each element with 32 zeros." (see col. 15, lines 38-40). Agarwal does not partition data to obtain more than one data elements in a register. In addition, Agarwal does not teach dynamically partitioning data based on an elemental width to obtain multiple operands. Furthermore, Agarwal does not teach performing group floating-point operations on multiple operands that are derived from dynamically partitioned data. That is, Agarwal does not teach a "multi-precision execution unit ... configurable to dynamically partition data received from the data path to account for an elemental width ... wherein the elemental width of the data is equal or narrower than the data path, the multi-precision execution unit being capable of performing group floating-point operations on multiple operands in partitioned fields of operand registers ...", as recited in claim 1.

It is well-settled that a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). Since Agarwal fails to disclose and teach above discussed features, as recited in claim 1, the Applicants respectfully submit that Agarwal does not anticipate claim 1. Thus, claim 1 is patentable. Therefore, the Applicants respectfully request that the rejection of claim 1 under 35 U.S.C. §102(e) be withdrawn.

Claims 2-30 depend from claim 1. Thus, claims 2-30 are patentable at least for the reasons stated above with respect to claim 1 and for the additional features recited therein. Therefore, the Applicant respectfully requests that the rejection of claims 2-30 under 35 U.S.C. §102(e) be withdrawn.

New Claims 31-47

The newly added claims 31-47 are fully supported by the specification as originally filed. Specifically, claim 31 is supported by Fig. 4 and paragraph 231 of the specification.

Claim 32-34, 35-37, 42-45 recite features related to execution of group floating-point arithmetic operations. Support for these claims can be found in Figs. 30A-30C as well as in paragraphs 269-272 of the specification.

Claims 34-35 and 38-40 recite features related to data handling operations. Support for these claims can be found in paragraphs 54 and 312-328 of the specification as well as the figures that are described in these paragraphs.

Claim 41 recites features related to a square root operation performed based on a group floating-point arithmetic instruction. Support for claim 41 can be found in Appendix, page 276-282.

Claim 46 recites features related to group integer arithmetic operations. Support for claim 46 can be found in paragraphs 262-266 of the specification as well as the figures that are described in these paragraphs.

Claim 47 recites features related to an execution unit. Support for claim 47 can be found at least in Fig. 1 where execution units 141-149 are shown as well as portions of the specification that describe the execution units.

New claims 31-34 depend from claim 1. Thus, claims 31-34 are patentable at least for the reasons stated above with respect to claim 1 and for the additional features recited therein.

New claims 35, 42, and 48 are independent claims. Claim 35 recites "a programmable processor ... in response to decoding a single instruction specifying an elemental width of operands and a floating-point arithmetic operation, the execution unit (i) partitions data received from the data path in an operand register based on the elemental width into a plurality of operands stored in partitioned fields of the operand register, (ii) performs the floating-point arithmetic operation on each of the plurality of operands ...". Claim 35 recites features including (1) dividing data from the data path width-wise into a variable number of data elements corresponding to partitioned fields, (2) that the division (or partition) is based upon an elemental width specified in the same instruction that specifies a floating-point arithmetic operation, and (3) that the partitioned fields correspond to multiple operands that are used in group floating-point operations.

As discussed above, Agarwal does not teach, disclose, or suggest that data in an operand register be partitioned into a variable number of data elements based on a dynamic elemental width specified in the same instruction for floating-point arithmetic operation. Therefore, the Applicants respectfully submit that Agarwal does not anticipate claim 35. Thus, claim 35 is patentable.

Claims 36-41 depend from claim 35. Thus, claims 36-41 are patentable at least for the reasons stated above with respect to claim 35 and for the additional features recited therein.

Claim 42 recites "a programmable microprocessor ... comprising ... an execution unit ... configurable to partition data, on an instruction-by-instruction basis, stored in an operand register having a width of n bits into a plurality of operands, each operand having an elemental width of m contiguous bits such that m times the number of operands in the plurality of operands equals n, the execution unit being capable of executing group floating-point arithmetic instructions ... on each of the plurality of operands to produce a plurality of individual m-bit results...". Claim 42 recites features, including (1) dividing data, on an instruction-by-instruction basis, width-wise into a variable number of data elements, each of which having an m-bits width, (2) that the division (or partition) is on an instruction-by-instruction basis based upon a dynamic elemental width, (3) each of the m-bits contiguous fields corresponds to an operand, and (4) individual operands are used in group floating-point operations to produce a plurality of corresponding m-bit results.

As discussed above, Agarwal does not teach, disclose, or suggest that data in an operand register be partitioned into a variable number of data elements based on a elemental width, specified on an instruction-by-instruction basis, and that such partitioned data elements are used in group floating-point operations to produce the same number of results. Therefore, the Applicants respectfully submit that Agarwal does not anticipate claim 42. Thus, claim 42 is patentable.

Claims 43-47 depend from claim 42. Thus, claims 43-47 are patentable at least for the reasons stated above with respect to claim 42 and for the additional features recited therein.

Claim 48 recites an instruction that causes a data processing system "dynamically partitioning the data stream based on an elemental width ... wherein the elemental width ... is

less than the width of the registers, and at least one of the instructions including group floating-point arithmetic instructions that specify one of a plurality of different elemental operand widths and a floating-point arithmetic operation and that perform the specified group floating-point arithmetic operation on multiple operands of the specified width ...". Claim 48 recites features including (1) dividing data from the data path width-wise into a variable number of data elements corresponding to multiple operands, (2) that the division (or partition) is based upon an elemental width which is dynamically specified in the same instruction specifying a group floating-point arithmetic operation, and (3) that the partitioned fields correspond to multiple operands that are used in group floating-point operations.

As discussed above, Agarwal does not teach, disclose, or suggest that data in an operand register be dynamically partitioned into a variable number of data elements based on an elemental width. Agarwal does not teach performing a group floating-point operation on multiple operands of a dynamically specified elemental width. Therefore, the Applicants respectfully submit that Agarwal does not anticipate claim 48. Thus, claim 48 is patentable.

Accordingly, it is believed that all pending claims are now in condition for allowance. Applicant therefore respectfully requests an early and favorable reconsideration and allowance of this application. If there are any outstanding issues which might be resolved by an interview or an Examiner's amendment, the Examiner is invited to call Applicant's representative at the telephone number shown below.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 500417 and please credit any excess fees to such deposit account.

Respectfully submitted,

McDERMOTT WILL & EMERY LLP

Kenneth L. Cage
Reg. No. 36,139
Registration No. 26,151

Please recognize our Customer No. 20277 as our correspondence address.

600 13th Street, N.W.
Washington, DC 20005-3096
Phone: 202.756.8000 KLC/QH:llg
Facsimile: 202.756.8087
Date: November 9, 2005

WDC99 1155167-1.043876.0145